

CLAIMS

Now, therefore, the following is claimed:

- 1 1. A system for enabling selective assertion testing of computer programs
2 based on run time inputs, comprising:
3 memory; and
4 a compiler configured to translate a first function of a first program stored in
5 said memory into a second function of a second program, said first function having
6 assertion instructions, said second function having translated assertion instructions
7 translated from said assertion instructions of said first function, said compiler
8 configured to enable selective execution of a portion of said translated assertion
9 instructions based on a run time input.
- 1 2. The system of claim 1, wherein said compiler is further configured to
2 enable the execution of each said translated assertion instruction that extends an
3 execution time of said second function to be prevented.
- 1 3. The system of claim 1, wherein said compiler is further configured to
2 enable selective execution of assertion handling code based on said run time input in
3 response to a failed assertion test that is defined by said translated assertion
4 instructions.

1 4. The system of claim 1, wherein said compiler is further configured to
2 detect stall locations in said second function and to insert into said stall locations each
3 said translated assertion instruction that is outside of said portion.

1 5. The system of claim 4, wherein said compiler is further configured to
2 enable execution of each said translated assertion instruction that is outside of said
3 portion regardless of said run time input.

1 6. A system for enabling selective assertion testing of computer programs
2 based on run time inputs, comprising:
3 memory; and
4 a compiler configured to translate a first program stored in said memory into a
5 second program, said first program having assertion instructions, said compiler
6 configured to translate said assertion instructions into translated assertion instructions
7 and to insert said translated assertion instructions into said second program, said
8 compiler configured to detect a stall location in said second program and to insert one
9 of said translated assertion instructions into said second program at said stall location
10 in response to a detection of said stall location, said compiler further configured to
11 insert mode selection instructions and mode test instructions in said second program,
12 wherein said mode selection instructions, when executed by a processing
13 element, cause said processing element to control a value of a mode indicator based
14 on a run time input, and wherein said mode test instructions, when executed by said
15 processing element, cause said processing element, based on said mode indicator, to
16 control whether at least one of said translated assertion instructions is executed by said
17 processing element.

1 7. The system of claim 6, wherein said compiler is configured to
2 automatically generate said mode selection instructions and said mode testing
3 instructions in response to at least one of said assertion instructions in said first
4 program.

1 8. The system of claim 6, wherein:
2 said translated assertion instructions and said mode testing instructions are
3 inserted into a single function of said second program by said compiler;
4 said compiler further is further configured to insert assertion handling
5 instructions and a second set of mode testing assertions into said second program and
6 outside of said function, said assertion handling instructions, when executed by said
7 processing element, cause said processing element to terminate execution of said
8 second program;
9 at least one of said translated assertion instructions, when executed by said
10 processing element, causes said processing element to branch to said second set of
11 mode testing assertions in response to a failed assertion test; and
12 said second set of mode test instructions, when executed by said processing
13 element, causes said processing element to control whether said assertion handling
14 instructions are executed based on said value of said mode indicator.

1 9. The system of claim 8, wherein said second set of mode test
2 instructions, when executed by said processing element, causes said processing
3 element to branch back to said function if said value of said mode indicator indicates
4 that said assertion handling instructions are not to be executed.

1 10. A method for enabling selective assertion testing of computer programs
2 based on run time inputs, comprising the steps of:
3 translating a first function of a first computer program into a second function
4 of a second computer program, said first function having assertion instructions, said
5 second function having translated assertion instructions translated from said assertion
6 instructions of said first function;
7 detecting stall locations within said second function of said second computer
8 program;
9 inserting one of said translated assertion instructions into one of said stall
10 locations in response to a detection of said one stall location in said detecting step;
11 inserting a block of said translated assertion instructions into said second
12 function; and
13 enabling selective execution of said block of translated assertion instructions
14 based on a run time input.

1 11. The method of claim 10, further comprising the step of executing said
2 one assertion instruction regardless of said run time input.

1 12. The method of claim 10, further comprising the step of:
2 enabling selective execution, based on said run time input, of assertion
3 handling code in response to a failed assertion test that is defined by said translated
4 assertion instructions.

1 13. The method of claim 10, further comprising the step of:
2 enabling execution of each said translated assertion instruction that is outside
3 of said block of translated assertion instructions regardless of said run time input.

1 14. A method for enabling selective assertion testing of computer programs
2 based on run time inputs, comprising the steps of:
3 translating a first function of a first computer program into a second function
4 of a second computer program, said first function having assertion instructions, said
5 second function having translated assertion instructions translated from said assertion
6 instructions of said first function; and
7 enabling selective execution of a portion of said translated assertion
8 instructions based on a run time input.

1 15. The method of claim 14, further comprising the step of preventing
2 execution of each said translated assertion instruction that extends an execution time
3 of said second function.

1 16. The method of claim 14, wherein said enabling step is automatically
2 performed along with said translating step.

1 17. The method of claim 14, further comprising the steps of:
2 detecting stall locations in said second function; and
3 inserting into said stall locations each said translated assertion instruction that
4 is outside of said portion.

1 18. The method of claim 17, further comprising the step of:
2 enabling execution of each said translated assertion instruction that is outside
3 of said portion regardless of said run time input.

1 19. The method of claim 17, further comprising the step of:
2 enabling selective execution, based on said run time input, of assertion
3 handling code in response to a failed assertion test defined by said translated assertion
4 instructions.